Magento 2 Installation with Nginx Webserver and Certbot SSL Certificate

1. Install Ubuntu and set date using dpkg-reconfigure tzdata
2. Install Nginx
   1. apt-get update
   2. apt-get -y install nginx
   3. to test nginx, open browser and open the ip address of web server
   4. create a new file with name superstore in /etc/nginx/sites-available and enter below code in it

upstream fastcgi\_backend {

server unix:/run/php/php7.2-fpm.sock;

}

server {

server\_name superstore.com 192.168.33.10;

listen 80;

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

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

}

* 1. create a symbolic link using below command

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

* 1. restart the nginx and php service

systemctl restart nginx && systemctl restart php7.2-fpm

* 1. test the nginx server by running command

nginx -t

nginx -T

1. Install PHP7.2-FPM , PHP-CLI and Related Modules
   1. apt-get -y install php7.2-fpm php7.2-cli or php-fpm in 20.04 ubuntu
   2. install the below php extensions
      1. sudo apt-get install -y php7.2-common php7.2-gd php7.2-mysql php7.2-curl php7.2-intl php7.2-xsl php7.2-mbstring php7.2-zip php7.2-bcmath php7.2-iconv php7.2-soap zip unzip php7.2-zip
   3. to setup php7.2 as default php version, run below command

update-alternatives --set php /usr/bin/php7.2

* 1. open /etc/php/7.2/fpm/php.ini and /etc/php/7.2/cli/php.ini and check below
     1. memory\_limit = 2G
     2. max\_execution\_time = 1800
     3. zlib.output\_compression = On
     4. opcache.enable=1
     5. opcache.save\_comments=yes
     6. date.timezone = Asia/Dubai
     7. upload\_max\_filesize = 100M
  2. restart php-fpm services
     1. systemctl restart php7.2-fpm

1. Install SQL Server
   1. apt-get install -y mysql-server mysql-client
   2. mysql\_secure\_installation (choose all Yes Option)
   3. if we want to import large data in mysq, edit sql.cnf file
   4. create magento database using below commands
      1. login to mysql
      2. create database database\_name;
      3. create user ‘username’@’localhost’ identified by ‘password’;
      4. grant all on database\_name.\* TO ‘username’@’localhost’ with grant option;
      5. flush privileges;
2. Installing Elasticsearch
   1. apt-get -y update && apt-get install -y openjdk-8-jdk

use above command to install java

* 1. java -version

use above command to check java version

* 1. sudo apt -y install gnupg
  2. wget -qO - <https://artifacts.elastic.co/GPG-KEY-elasticsearch> | sudo apt-key add -
  3. sudo apt -y install apt-transport-https
  4. echo "deb <https://artifacts.elastic.co/packages/oss-7.x/apt> stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
  5. sudo apt update
  6. sudo apt -y install elasticsearch-oss
  7. sudo systemctl daemon-reload
  8. sudo systemctl enable elasticsearch.service
  9. sudo systemctl start elasticsearch.service
  10. sudo systemctl status elasticsearch.service

use above command from c to l to install elasticsearch

j. curl -XGET 'localhost:9200/\_cat/health?v&pretty'

use above command to check & verify elasticsearch installation

1. Install Magento Using Composer method
   1. Prerequisites
      1. System Requirements
         1. Ubutnu
         2. 2GB RAM
         3. PHP & MYSQL , NGINX, COMPOSER, PHP Required extensions, SSL,
      2. Magento file system owner
      3. Install composer
      4. apt install curl git

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

* + 1. Obtain authentication keys

Username = 6d304bfc5ac8e8cdac4c60c14247ca39

Password = 65efe00a6ff3fcaf29d8ac13aef70531

Run the below command from magento directive to install magento.

sudo bin/magento setup:install --base-url-secure=https://example.com/ --db-host=localhost --db-name=magento --db-user=magento --db-password=m@g1t786 --admin-firstname=rizwan --admin-lastname=Khan --admin-email=Rizwan.wakil@gmail.com --admin-user=rizwan --admin-password=admin123 --language=en\_US --currency=USD --timezone=Asia/Dubai --use-rewrites=1

To disable two factor authentication, issue below command

bin/magento mo:di Magento\_TwoFactorAuth

* 1. Creating User and Permission setup
     1. Create user using command below
        1. adduser username
        2. passwd username
        3. usermod -a -G www-data username
  2. Install Composer
     1. sudo apt-get install curl git
     2. sudo curl -sS https://getcomposer.org/installer | php
     3. sudo mv composer.phar /usr/local/bin/composer
     4. or use the below command as per magento website

curl -sS https://getcomposer.org/installer | sudo php -

- --install-dir=/usr/bin --filename=composer

* + 1. Get the magento software using composer

1. Create a directory with name superstore
   * 1. Move superstore directory and issue below command

composer create-project --repository-url=https://repo.m

agento.com/ magento/project-community-edition .j

* + 1. Now run the setup wizard from browser by type server\_ip\_address

1. Setup Permission for Magento files

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

1. Installing Sample Data in Magento
   1. php bin/magento sampledata:deploy
   2. php bin/magento setup:upgrade
2. Creating Magento Crontab
   1. Change to magento root directory
   2. Enter below command

php bin/magento crontab:install –force

* 1. To view crontab, enter following command

Crontab -l

1. Installing Extension
   1. Backup your database
   2. Enable maintenance mode using below command

php bin/magento maintenance:enable

* 1. Get extension from magento marketplace
  2. Get extension composer name & version by click profile 🡪 My Purchase 🡪 Technical Detail
  3. Update composer.json file in magento project with name &verion of extension by running below command from magento root directory

Composer require extension\_name:version

* 1. Verify extension install properly by running below command

php bin/magento module:status extension\_name

* 1. Enable and configure the extension

To enable the module run below command

php bin/magento module:enable module\_name

To register the module run below command

php bin/magento setup:upgrade

Then run the magent recompile command

php bin/magento setup:di:compile

1. Upgrade an Extension
   1. Copy the composer file to magento root
   2. Run below command to update composer

Composer update vendor/module\_name

Composer require vendor/module-name ^x.x.x

* 1. Run below commands

php bin/magento setup:upgrade –keep-generated

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

php bin/magento cache:clean

1. Installing phpMyAdmin
   1. Run below command to install phpMyAdmin

apt-get install phpMyAdmin

* 1. To access the phpMyAdmin, enter below url in browser

192.168.33.10/phpMyAdmin

1. SSL Installation & Configuration

apt update

install certbot using below command

add-apt-repository ppa:certbot/certbot

install certbot nginx package

apt install python-certbot-nginx

obtain SSL certificate using certbot

certbot –nginx -d superstore.com

above command will work for valid domain name only

your certificate will be generated and stored in

/etc/letsencrypt/live/yourdomainname/fullchain.pem

Add the following line below the nginx configuration file

listen 443 ssl;

ssl\_certificate /etc/letsencrypt/live/yourdomain.com/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/yourdomain.com/privkey.pem;

include /etc/letsencrypt/options-ssl-nginx.conf;

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem;

now enable SSL in magento 2 from admin

store🡪 configuration 🡪 General 🡪 web 🡪 Base URLs (Secure)

set URL and enable it.

If we are not able to access our admin panel, we can set using command line

php bin/magento setup:store-config:set –use-secure-admin=1

1. Install Varnish and Configuration
   1. Run below command to install and start and enable varnish

apt-get install varnish

systemctl start varnish

systemctl enable varnish

* 1. By default varnish listen on port 6081 for public and 6082 for backend
  2. Checking varnish service is running

netstat -plntu

* 1. Add the following line below the nginx configuration file