

# Documentation for work with MAGENTO 2.4.4 (in docker-compose)

In this example was used:

NAME="Ubuntu"

VERSION="20.04.4 LTS (Focal Fossa)"

```
root@magento:/home/hesher# cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.4 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.4 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
root@magento:/home/hesher#
```

Docker version 20.10.14, build a224086349

```
root@magento:/home/hesher# docker --version
Docker version 20.10.14, build a224086349
root@magento:/home/hesher#
```

Detailed infrastructure can be viewed here:

<https://github.com/yevheniistelmakh/Mag2>

Create work folder and inside create next sub-folders:

-nginx

-php-fpm

-magento2 (source code)

```
root@magento:/home/hesher/MAGENT0# ls -al
total 28
drwxr-xr-x  6 root   root   4096 May 14 11:15 .
drwxr-xr-x 15 hesher hesher 4096 May 14 11:15 ..
-rw-r--r--  1 root   root   2429 Apr 26 14:25 docker-compose.yml
drwxr-xr-x  8 root   root   4096 Apr 17 17:42 .git
drwxr-xr-x 13 root   root   4096 Apr 17 13:02 magento2
drwxr-xr-x  3 root   root   4096 May 13 08:05 nginx
drwxr-xr-x  3 root   root   4096 Apr 28 15:55 php-fpm
root@magento:/home/hesher/MAGENT0#
```

In the folder “nginx” we created a sub-folder “config” and Dockerfile for nginx image.

In the dockerfile user “www-data” should be created . The user ID and epy name must match the user that was created in both the php image and the nginx image. This user is needed so that nginx can access the php container.

```
drwxr-xr-x 3 root root 4096 May 14 11:27 .
drwxr-xr-x 6 root root 4096 May 14 11:15 ..
drwxr-xr-x 4 root root 4096 Apr 17 15:41 config
-rw-r--r-- 1 root root 132 Apr 17 16:29 Dockerfile
root@magento:/home/hesher/MAGENT0/nginx#
```

\*Dockerfile for nginx

FROM nginx:1.21.6-alpine

RUN adduser -D -H -u 1005 -s /bin/bash -G www-data www-data

COPY ./config/templates/ /etc/nginx/templates/

```
FROM nginx:1.21.6-alpine
RUN adduser -D -H -u 1005 -s /bin/bash -G www-data www-data
COPY ./config/templates/ /etc/nginx/templates/
```

Next create 2 sub-folders in “config” folder

```
root@magento:/home/hesher/MAGENT0/nginx/config# ls -al
total 16
drwxr-xr-x 4 root root 4096 Apr 17 15:41 .
drwxr-xr-x 3 root root 4096 May 14 11:27 ..
drwxr-xr-x 2 root root 4096 Apr 15 16:59 files
drwxr-xr-x 2 root root 4096 May 13 16:16 templates
root@magento:/home/hesher/MAGENT0/nginx/config#
```

- files (Add index.html file to this folder)

```
root@magento:/home/hesher/MAGENT0/nginx/config/files# cat index.html
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8">
<title>Урок HTML разметки</title>
<meta name="description" content="hi all">
<meta name="keywords" content="How are you?">
</head>
<body>

</body>
</html>
root@magento:/home/hesher/MAGENT0/nginx/config/files#
```

- templates (Here magento.conf.template was created, which, when the container starts, will pick it up and create the necessary configuration file in nginx container /etc/nginx/conf.d/magento.conf.)

<https://github.com/yevheniistelmakh/Mag2/blob/master/nginx/config/templates/magento.conf.template>

```

# Example configuration:
upstream fastcgi_backend {
    # use tcp connection
    server ${PHP_FPM_HOST}:${PHP_FPM_PORT};
    # or socket
    # server unix:/var/run/php/php7.0-fpm.sock;
}
server {
    server_name _;
    listen 80 default_server;
    set $MAGE_ROOT /var/www/html;
    set $MAGE_DEBUG_SHOW_ARGS 1;
    client_max_body_size 100M;
    # include /vagrant/magento2/nginx.conf.sample;

    #
    ## Optional override of deployment mode. We recommend you use the
    ## command 'bin/magento deploy:mode:set' to switch modes instead.
    ##
    ## set $MAGE_MODE default; # or production or developer
    ##
    ## If you set MAGE_MODE in server config, you must pass the variable into the
    ## PHP entry point blocks, which are indicated below. You can pass
    ## it in using:
    ##
    ## fastcgi_param MAGE_MODE $MAGE_MODE;
    ##
    ## In production mode, you should uncomment the 'expires' directive in the /static/ location block

    root $MAGE_ROOT/pub;

    index index.php;
    autoindex off;
    charset UTF-8;
    error_page 404 03 = /errors/404.php;
    #add_header "X-UA-Compatible" "IE=Edge";

```

Pay attention to one nuance in nginx.conf at **server\_name \_;** .

```

# Example configuration:
upstream fastcgi_backend {
    # use tcp connection
    server ${PHP_FPM_HOST}:${PHP_FPM_PORT};
    # or socket
    # server unix:/var/run/php/php7.0-fpm.sock;
}
server {
    server_name _;
    listen 80 default_server;
    set $MAGE_ROOT /var/www/html;
    set $MAGE_DEBUG_SHOW_ARGS 1;
    client_max_body_size 100M;
    # include /vagrant/magento2/nginx.conf.sample;

    #
    ## Optional override of deployment mode. We recommend you use the

```

Next step will be to create a php folder.

```

drwxr-xr-x 6 root root 4096 May 14 12:50 .
drwxr-xr-x 15 hesher hesher 4096 May 14 11:15 ..
-rw-r--r-- 1 root root 2429 Apr 26 14:25 docker-compose.yml
drwxr-xr-x 8 root root 4096 Apr 17 17:42 .git
drwxr-xr-x 13 root root 4096 Apr 17 13:02 magento2
drwxr-xr-x 3 root root 4096 May 14 11:27 nginx
drwxr-xr-x 3 root root 4096 Apr 28 15:55 php-fpm
root@magento:/home/hesher/MAGENTO#

```

After we need to create a folder files and a Dockerfile for the php image.

```

drwxr-xr-x 3 root root 4096 Apr 28 15:55 .
drwxr-xr-x 6 root root 4096 May 14 12:50 ..
-rw-r--r-- 1 root root 1092 Apr 26 13:47 Dockerfile
drwxr-xr-x 2 root root 4096 Apr 15 20:20 files
root@magento:/home/hesher/MAGENTO/php-fpm#

```

In the Dockerfile, specify to create a user www-data and install php dependencies.

```
FROM php:8.1-fpm-alpine3.14

RUN deluser www-data && adduser -u 1005 --disabled-password www-data

# RUN apk add --update --no-cache zlib-dev libpng-dev freetype-dev libjpeg-turbo-dev curl-dev aws-cli\
# icu-dev oniguruma-dev libxslt-dev libzip-dev \
# && docker-php-ext-configure gd \
# --with-freetype-dir=/usr/lib/ \
# --with-png-dir=/usr/lib/ \
# --with-jpeg-dir=/usr/lib/ \
# --with-gd \
# && docker-php-ext-install -j$(nproc) gd \
# && docker-php-ext-install -j$(nproc) bcmath intl pdo_mysql soap zip xsl sockets

RUN curl -sSLf \
-o /usr/local/bin/install-php-extensions \
https://github.com/mlocati/docker-php-extension-installer/releases/latest/download/install-php-extensions && \
chmod +x /usr/local/bin/install-php-extensions && \
install-php-extensions gd

RUN install-php-extensions bcmath intl pdo_mysql soap sockets xsl zip
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/bin --filename=composer
#50OPY ./magento2 /var/www/html
COPY ./files/php.ini "${PHP_INI_DIR}/php.ini"
```

In folder files create php.ini file.

Example:

<https://github.com/yevheniistelmakh/Mag2/blob/master/php-fpm/files/php.ini>

```
drwxr-xr-x 2 root root 4096 Apr 15 20:20 .
drwxr-xr-x 3 root root 4096 May 14 13:19 ..
-rw-r--r-- 1 root root 71554 Apr 15 20:20 php.ini
root@magento:/home/hesher/MAGENTO/php-fpm/files#
```

In this file change memory limit to 2Gb (Required MINIMUM 2Gb)

```
; Maximum amount of memory a script may consume (128MB)
; http://php.net/memory-limit
memory_limit = 2048M
```

Create a magento folder.

```
drwxr-xr-x 6 root root 4096 May 14 12:50 .
drwxr-xr-x 15 hesher hesher 4096 May 14 11:15 ..
-rw-r--r-- 1 root root 2429 Apr 26 14:25 docker-compose.yml
drwxr-xr-x 8 root root 4096 Apr 17 17:42 .git
drwxr-xr-x 13 root root 4096 Apr 17 13:02 magento2
drwxr-xr-x 3 root root 4096 May 14 11:27 nginx
drwxr-xr-x 3 root root 4096 May 14 13:19 php-fpm
root@magento:/home/hesher/MAGENTO#
```

Now download the source code Magento from Github.

(<https://github.com/magento/magento2>)

The screenshot displays the GitHub interface for the `magento/magento2` repository. At the top, it shows the repository name, 25 branches, and 148 tags. Below this, there's a 'Switch branches/tags' dropdown menu with '2.4-develop' selected. The main area shows a list of recent commits, including 'magento-performance/MCP-954', 'ENGCOM-9112: Update CONTRIBUTING.md #33130', and 'Merge remote-tracking branch 'origin/2.4-develop' into MCP-954'. The right sidebar contains the 'About' section, which describes the repository as the source code for Magento 2, and provides links to the website and other resources.

cd magento2  
git clone <https://github.com/magento/magento2.git>  
Execute "git checkout" and select the 2.4.4 version of magento.

## Creation of docker-compose.yml

<https://github.com/yevheniistelmakh/Mag2/blob/master/docker-compose.yml>

```
total 28  
drwxr-xr-x  6 root    root    4096 May 14 11:15 .  
drwxr-xr-x 15 hesher  hesher 4096 May 14 11:15 ..  
-rw-r--r--  1 root    root   2429 Apr 26 14:25 docker-compose.yml  
drwxr-xr-x  8 root    root    4096 Apr 17 17:42 .git  
drwxr-xr-x 13 root    root    4096 Apr 17 13:02 magento2  
drwxr-xr-x  3 root    root    4096 May 14 11:27 nginx  
drwxr-xr-x  3 root    root    4096 Apr 28 15:55 php-fpm  
root@magento:/home/hesher/MAGENTO#
```

version: '3.7'

services:

sigma\_magento-nginx:

container\_name: sigma\_magento-nginx

volumes:

- ./magento2:/var/www/html

build:

context: ./nginx/

ports:

- 8081:80

restart: always

environment:

- PHP\_FPM\_HOST=sigma\_magento-php\_fpm

- PHP\_FPM\_PORT=9000

sigma\_magento-php\_fpm:

build:

context: ./php-fpm/

container\_name: sigma\_magento-php\_fpm

volumes:

- ./magento2:/var/www/html

restart: always

sigma\_magento\_db:

image: mysql:8.0-debian

restart: always

command: --default-authentication-plugin=mysql\_native\_password

container\_name: sigma\_magento\_db

environment:

MYSQL\_DATABASE: 'local\_magento\_db'

```
# So you don't have to use root, but you can if you like
MYSQL_USER: 'local_agent'
# You can use whatever password you like
MYSQL_PASSWORD: 'local_password'
# Password for root access
MYSQL_ROOT_PASSWORD: 'root_strong_password'
ports:
# <Port exposed> : < MySQL Port running inside container>
- '3306:3306'
expose:
# Opens port 3306 on the container
- '3306'
# Where our data will be persisted
volumes:
- my-db:/var/lib/mysql
```

```
# Elasticsearch Docker Images: https://www.docker.elastic.co/
sigma_magento_elasticsearch:
image: elasticsearch:7.16.3
container_name: sigma_magento_elasticsearch
environment:
- xpack.security.enabled=false
- discovery.type=single-node
ulimits:
memlock:
soft: -1
hard: -1
nofile:
soft: 65536
hard: 65536
cap_add:
- IPC_LOCK
volumes:
- elasticsearch-data:/usr/share/elasticsearch/data
ports:
- 9200:9200
- 9300:9300
```

```
# Names our volume
volumes:
my-db:
elasticsearch-data:
driver: local
```

## Run docker-compose

Please note that we use the following Database (mysql:8.0-debian) and Elasticsearch (elasticsearch:7.16.3). These requirements are written in the

[Magento documentation](#).

<https://devdocs.magento.com/guides/v2.4/install-gde/system-requirements.html>)

Next run docker compose.

run:

```
docker-compose up -d
```

```
docker ps
```

Go into the php container and go to the /var/www/html/ folder and run the command:

```
composer instal
```

```
composer -V
```

The next step is very important. We install Magento

Please see more information about installing Magento here:

<https://devdocs.magento.com/guides/v2.4/install-gde/install/cli/install-cli.html>

```
bin/magento setup:install \
  --base-url=http://magento.local:8081 \
  --db-host=sigma_magento_db \
  --db-name=local_magento_db \
  --db-user=local_agent \
  --db-password=local_password \
  --admin-firstname=Magento \
  --admin-lastname=User \
  --admin-email=stelmakh.yevhenii.i@gmail.com \
  --admin-user=admin \
  --admin-password=admin123 \
  --language=en_US \
  --currency=USD \
  --timezone=America/Chicago \
  --use-rewrites=1 \
  --search-engine=elasticsearch7 \
  --elasticsearch-host=sigma_magento_elasticsearch \
  --elasticsearch-port=9200
```

At the end you should see a link to the admin panel that you need to add to our domain.name:8081/admin\_...

```

Module 'Magento_WishlistAnalytics':
[Progress: 844 / 851]
Module 'Magento_WishlistGraphQL':
[Progress: 845 / 851]
Enabling caches:
Current status:
layout: 1
block_html: 1
full_page: 1
[Progress: 846 / 851]
Installing admin user...
[Progress: 847 / 851]
Caches clearing:
Cache cleared successfully
[Progress: 848 / 851]
Disabling Maintenance Mode:
[Progress: 849 / 851]
Post installation file permissions check...
For security, remove write permissions from these directories: '/var/www/html/app/etc'
[Progress: 850 / 851]
Write installation date...
[Progress: 851 / 851]
[SUCCESS]: Magento installation complete.
[SUCCESS]: Magento Admin URI: /admin_1hvicr
Nothing to import.

```

After all manipulations, it is better to check the rights to the folder with Magento (/var/www/html/) and which user is the owner (should be www-data (this is the user we created when building the PHP and nginx images)  
 RUN `adduser -D -H -u 1005 -s /bin/bash -G www-data www-data`)

To change the user, use the command in the php container  
`chown -R www-data /var/www/html/`

After that, go to the browser using the domain name that you specified in the Magento install

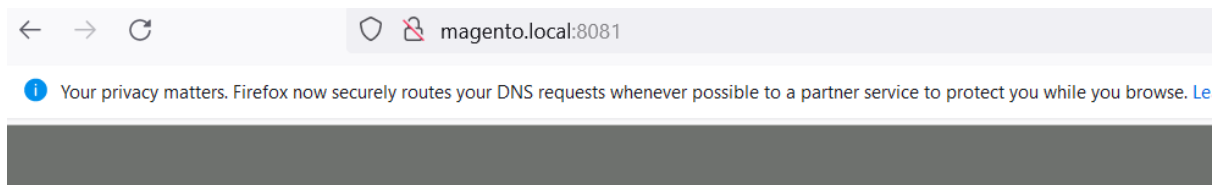
```

bin/magento setup:install \
--base-url=http://magento.lolchto.pp.ua \
--db-host=magento-db.chulijqohino.eu-central-1.rds.amazonaws.com \
--db-name=local_magento_db \
--db-user=admin \
--db-password=adminadmin \
--admin-firstname=admin \
--admin-lastname=admin \
--admin-email=stelmakh.yevhenii.i@gmail.com \
--admin-user=admin \
--admin-password=Qwertv1 \

```

If everything went well, then you should see the Magento start page.

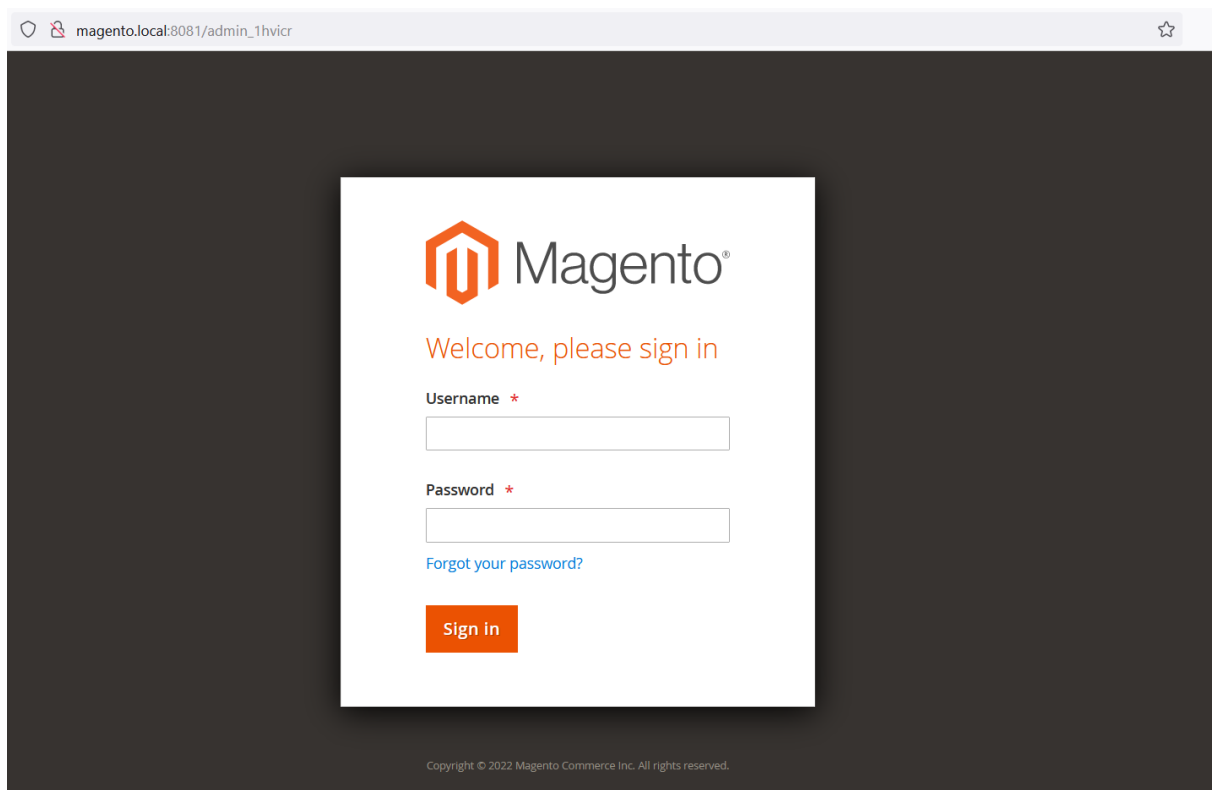




To access the admin panel use the admin address that you received after the successful installation of Magento (as an example:

[http://magento.local:8081/admin\\_1hvicr](http://magento.local:8081/admin_1hvicr) )

By clicking the link you should see the admin-panel:



Use the username and password that you specified when Magento was installing

```
bin/magento setup:install \
--base-url=http://magento.lolchto.pp.ua \
--db-host=magento-db.chulijqohino.eu-central-1.rds.amazonaws.com \
--db-name=local_magento_db \
--db-user=admin \
--db-password=adminadmin \
--admin-firstname=admin \
--admin-lastname=admin \
--admin-email=stelmakh.yevhenii.i@gmail.com \
--admin-user=admin \
--admin-password=Qwerty1 \
--language=en_US \
--currency=USD \
```

Dashboard / Magento Admin

← → ↺

🔒 magento.local:8081/admin\_1hvicr/admin/dashboard/index/key/c6bf18bfc87ae226f93b932f6d33fa946f2d3bcbef9da0f213aaec4f4fc4c35a/

☆

🔒

⚠️ One or more **indexers** are invalid. Make sure your **Magento cron job** is running. System Messages: 1

DASHBOARD

SALES

CATALOG

CUSTOMERS

MARKETING

CONTENT

REPORTS

STORES

SYSTEM

FIND PARTNERS & EXTENSIONS

## Dashboard

🔍 🔔 👤 admin

Scope: All Store Views ? Reload Data

### Advanced Reporting

Gain new insights and take command of your business' performance, using our dynamic product, order, and customer reports tailored to your customer data. Go to Advanced Reporting

#### Lifetime Sales

**\$0.00**

Chart is disabled. To enable the chart, click [here](#).

Revenue	Tax	Shipping	Quantity
<b>\$0.00</b>	<b>\$0.00</b>	<b>\$0.00</b>	<b>0</b>

#### Average Order

**\$0.00**

#### Last Orders

We couldn't find any records.

#### Last Search Terms

We couldn't find any records.

#### Top Search Terms

Bestsellers

Most Viewed Products

New Customers

Customers

We couldn't find any records.

## Changes to settings and login details

### Database

To change the Database parameters, go to the php container. Go to the /var/www/html/app/etc/env.php directory and find the env.php file.

There you will find the Database settings and a link to the Magento admin panel.

```
/var/www/html/app/etc # cat env.php
<?php
return [
    'queue' => [
        'consumers_wait_for_messages' => 1
    ],
    'backend' => [
        'frontName' => 'admin_1hvicr'
    ],
    'remote_storage' => [
        'driver' => 'file'
    ],
    'crypt' => [
        'key' => 'f94703f58bb66466ab1768777ba8cc94'
    ],
    'db' => [
        'table_prefix' => '',
        'connection' => [
            'default' => [
                'host' => 'sigma_magento_db',
                'dbname' => 'local_magento_db',
                'username' => 'local_agent',
                'password' => 'local_password',
                'model' => 'mysql4',
                'engine' => 'innodb',
                'initStatements' => 'SET NAMES utf8;',
                'active' => '1',
                'driver_options' => [
                    1014 => false
                ]
            ]
        ]
    ]
]
```

### Elasticsearch/OpenSearch

You can learn more about changes in Elasticsearch parameters here:

<https://devdocs.magento.com/guides/v2.4/config-guide/elasticsearch/configure-magento.html>

#### Configure your search engine from the Admin

To configure your system to use Elasticsearch or OpenSearch:

1. Log in to the Admin as an administrator.
2. Click Stores > Settings > Configuration > Catalog > Catalog Search.
3. From the Search Engine list, select the corresponding version of your search engine. If you are using OpenSearch, you must select Elasticsearch7.

The following table lists the required configuration options to configure and test the connection with Magento. Unless you changed the server settings of your search engine, the defaults should work. Skip to the next step.

NEW TEMPLATE  
добавить ссылки красивое.